New Retrieval Approaches Using SMART : TREC 4

Chris Buckley*, Amit Singhal, Mandar Mitra, (Gerard Salton)

Abstract

The Smart information retrieval project emphasizes completely automatic approaches to the understanding and retrieval of large quantities of text. We continue our work in TREC 4, performing runs in the routing, ad-hoc, confused text, interactive, and foreign language environments.

Introduction

For over 30 years, the Smart project at Cornell University has been interested in the analysis, search, and retrieval of heterogeneous text databases, where the vocabulary is allowed to vary widely, and the subject matter is unrestricted. Such databases may include newspaper articles, newswire dispatches, textbooks, dictionaries, encyclopedias, manuals, magazine articles, and so on. The usual text analysis and text indexing approaches that are based on the use of thesauruses and other vocabulary control devices are difficult to apply in unrestricted text environments, because the word meanings are not stable in such circumstances and the interpretation varies depending on context. The applicability of more complex text analysis systems that are based on the construction of knowledge bases covering the detailed structure of particular subject areas, together with inference rules designed to derive relationships between the relevant concepts, is even more questionable in such cases. Complete theories of knowledge representation do not exist, and it is unclear what concepts, concept relationships, and inference rules may be needed to understand particular texts.[8]

Accordingly, a text analysis and retrieval component must necessarily be based primarily on a study of the available texts themselves. Fortunately very large text databases are now available in machine-readable form, and a substantial amount of information is automatically derivable about the occurrence properties of words and expressions in natural-language texts, and about the contexts in which the words are used. This information can help in determining whether a query and a text are semantically homogeneous, that is, whether they cover similar subject areas. When that is the case, the text can be retrieved in response to the query.

Automatic Indexing

In the Smart system, the vector-processing model of retrieval is used to transform both the available information requests as well as the stored documents into vectors of the form:

$$D_i = (w_{i1}, w_{i2}, \ldots, w_{it})$$

where D_i represents a document (or query) text and w_{ik} is the weight of term T_k in document D_i. A weight of zero is used for terms that are absent from a particular document, and positive weights characterize terms actually assigned. The assumption is that t terms in all are available for the representation of the information.

In choosing a term weighting system, low weights should be assigned to high-frequency terms that occur in many documents of a collection, and high weights to terms that are important in particular documents but unimportant in the remainder of the collection. The weight of terms that occur rarely in a collection is

*Department of Computer Science, Cornell University, Ithaca, NY 14853-7501. This study was supported in part by the National Science Foundation under grant IRI 93-00124.

relatively unimportant, because such terms contribute little to the needed similarity computation between different texts.

A well-known term weighting system following that prescription assigns weight w_{ik} to term T_k in query Q_i in proportion to the frequency of occurrence of the term in Q_i, and in inverse proportion to 'the number of documents to which the term is assigned.[9, 7] Such a weighting system is known as a tf×idf (term frequency times inverse document frequency) weighting system. In practice the query lengths, and hence the number of non-zero term weights assigned to a query, varies widely.

The terms T_k included in a given vector can in principle represent any entities assigned to a document for content identification. In the Smart context, such terms are derived by a text transformation of the following kind:[7]

1. recognize individual text words
2. use a stop list to eliminate unwanted function words
3. perform suffix removal to generate word stems
4. optionally use term grouping methods based on statistical word co-occurrence or word adjacency computations to form term phrases (alternatively syntactic analysis computations can be used)
5. assign term weights to all remaining word stems and/or phrase stems to form the term vector for all information items.

Once term vectors are available for all information items, all subsequent processing is based on term vector manipulations.

The fact that the indexing of both documents and queries is completely automatic means that the results obtained are reasonably collection independent and should be valid across a wide range of collections. No human expertise in the subject matter is required for either the initial collection creation, or the actual query formulation.

Phrases

The same phrase strategy (and phrases) used in all previous TRECs ([4, 2, 5]) are used for TREC 4. Any pair of adjacent non-stopwords is regarded as a potential phrase. The final list of phrases is composed of those pairs of words occurring in 25 or more documents of the initial TREC 1 document set. Phrase weighting is again a hybrid scheme where phrases are weighted with the same scheme as single terms, except that normalization of the entire vector is done by dividing by the length of the single term sub-vector only. In this way, the similarity contribution of the single terms is independent of the quantity or quality of the phrases.

Text Similarity Computation

When the text of document D_i is represented by a vectors of the form $(d_{i1}, d_{i2}, \ldots, d_{it})$ and query Q_j by the vector $(q_{j1}, q_{j2}, \ldots, q_{jt})$, a similarity ($S$) computation between the two items can conveniently be obtained as the inner product between corresponding weighted term vector as follows:

$$S(D_i, Q_j) = \sum_{k=1}^{t} (d_{ik} * q_{jk}) \tag{1}$$

Thus, the similarity between two texts (whether query or document) depends on the weights of coinciding terms in the two vectors.

System Description

The Cornell TREC experiments use the SMART Information Retrieval System, Version 12, and are run on a dedicated Sun Sparc 20/51 with 160 Megabytes of memory and 27 Gigabytes of local disk.

SMART Version 12 is the latest in a long line of experimental information retrieval systems, dating back over 30 years, developed under the guidance of G. Salton. The new version is approximately 44,000 lines of C code and documentation.

SMART Version 12 offers a basic framework for investigations of the vector space and related models of information retrieval. Documents are fully automatically indexed, with each document representation being a weighted vector of concepts, the weight indicating the importance of a concept to that particular document (as described above). The document representatives are stored on disk as an inverted file. Natural language queries undergo the same indexing process. The query representative vector is then compared with the indexed document representatives to arrive at a similarity (equation (1)), and the documents are then fully ranked by similarity.

Document length normalization

It has become increasingly obvious over the past two years that the standard SMART method of document length normalization, cosine normalization, does not work optimally in the TREC environment.

The cosine similarity function (or equivalently, cosine document normalization) was developed in an era in which documents were short, and about a single topic. It emphasizes the relationship between the query and the entire document. Negative information, the fact that large parts of the document are *not* related to the query, is just as important as positive information.

Use of negative information is no longer appropriate if there are longer, full text documents to be retrieved. These long documents will have several sub-topics, only one of which may be pertinent to a query. Normalization using the cosine function will make these documents very difficult to retrieve, since the negative information will dominate.

Figure 1 shows the mismatch between the probability of retrieval of cosine normalized documents of a given length, and the probability of relevance of documents of that length. In an ideal graph where the system was accurately retrieving documents independent of length effects, these two curves would be co-incident.

The documents used for the TREC 4 ad hoc task were sorted by length into 568 bins of 1000 documents each. For each TREC 4 query we retrieve 1000 documents using our standard cosine normalized "lnc.ltc" weighting function and analyze which buckets the retrieved documents occur in, and which buckets the relevant documents occur in. I.e. for each Bin_i, we calculate Prob(Bin_i — Relevant) and Prob(Bin_i — Retrieved) Those numbers are plotted as the y-axis of Graph 1, with the x-axis being the median length of the documents within the sample buckets.

As would be expected, the probability of relevance of a document increases with length. Longer documents have more of a chance of having a relevant sub-topic since they have more sub-topics. However, the probability of retrieval of our cosine normalized documents does not at all match this increase! In fact the probability of retrieval remains roughly constant up until a length of 3000 bytes and then starts decreasing. Thus, our "lnc.ltc" measure retrieves a much larger share of short documents than it should, and a much smaller number of long documents.

This bias towards short documents affects much more than just straightforward adhoc retrieval. Our work in massive query expansion and our local/global approach have been strongly influenced by the bias. Over half of the effectiveness increases of each of these two approaches are due to their indirectly overcoming this bias, and being able to retrieve long documents.

Our local/global matching has given us 15% improvement in past tests [5]. The local match on fixed size windows has been explicitly non-normalized. Thus short and long documents have been treated equally on the local match, making up for the biased global match. If a good, non-biased normalization approach is used for the global match, then the improvement due to our current local match is reduced to about 3%.

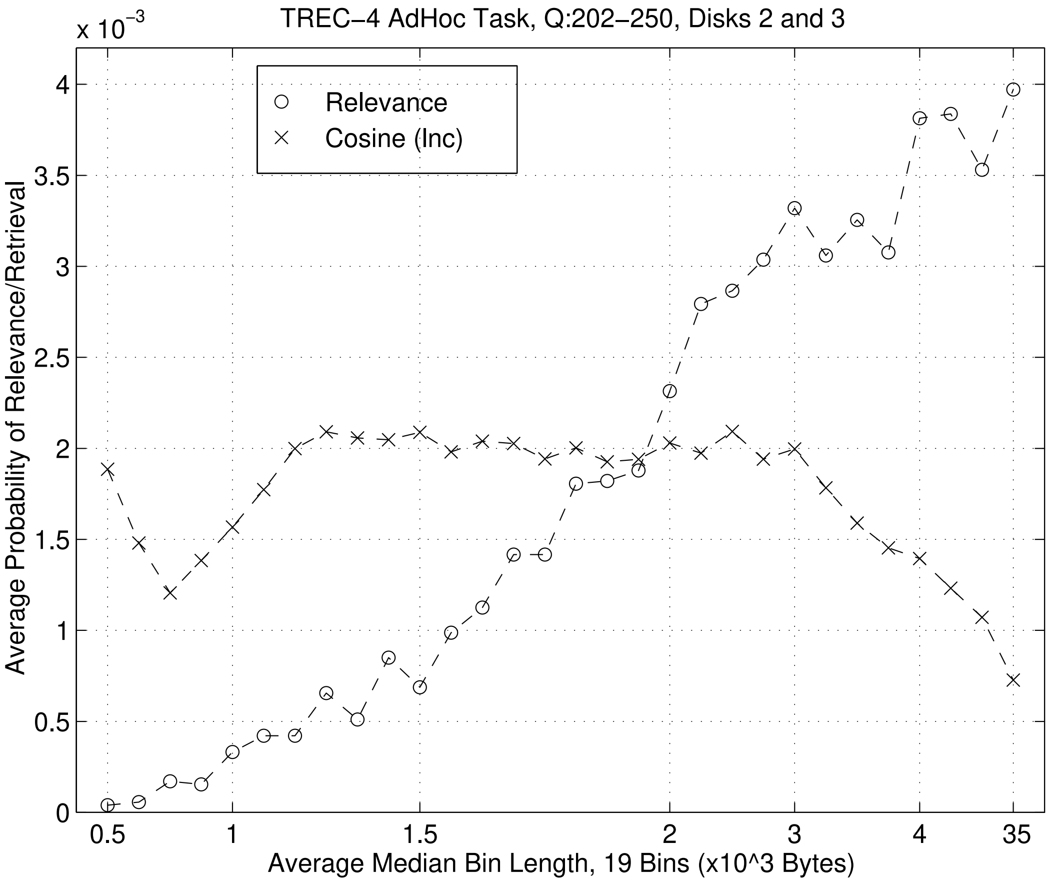

Figure 1: Probability of relevance and cosine retrieval for varying document length.

The effect on massive query expansion is a bit more subtle and harder to isolate. Consider the effect of adding 100 random (not related to relevance) common terms to a query. This will not have a large effect on a given short document since those terms are not likely to occur. A long document will be much more strongly affected since these terms will occur by "random chance". Thus long documents will have a higher comparative similarity than short documents due to these added terms, and we have an effect counter-balancing the short document bias of the cosine normalization. The end result has been an effective similarity approach, achieved through combining two biased approaches.

In our past work,[2, 5, 1], we've been expanding by 200 – 300 terms before reaching a point of diminishing returns. In contrast, starting with a non-biased normalization the maximum effect occurs at 80 – 100 added terms, and there is a slight decrease of 2% if 300 terms are added. For most of our test sets, the end result of the expansion is 10% – 15% better with the non-biased normalization; the expansion terms themselves do not help as much in the non-biased case, but the base non-expanded similarity starts off much better.

The background and derivation of the weighting function Cornell used in TREC 4 is described in Singhal et al [11]. Normalization is based upon both normalizing the *tf* factor by the average *tf* in the vector, and the overall vector length by a factor dependent on the number of unique terms. Based on this *tf* factor (which we call the L factor in Smart's term weight triple notation [9]) and pivoted unique normalization (which we call the u normalization), we obtain the final weighting strategy of the documents (called *Lnu* weighting in Smart):

$$\frac{\frac{1+log(tf)}{1+log(average\ \ tf)}}{(1.0 - slope) \times pivot + slope \times \#\ of\ \ unique\ \ terms}$$

Within our experiments here, *slope* is fixed at 0.2, and the pivot is set to the average number of unique terms occurring in the collection.

A major portion of our work since TREC 3 has been concerned with document length normalization [11], and the ramifications. The measure presented here will not be our final normalization measure, but it performs equivalently. We want the final measure to be independent of stemming and erroneous text like misspellings and OCR errors.

CrnlAE – Adhoc expansion run

The first of Cornell's two ad-hoc runs, CrnlAE, is very similar to our TREC 3 CrnlAE run. Last year, an initial retrieval was done, the top 30 documents were assumed to be relevant and submitted to our standard Rocchio relevance feedback procedure. The query was then expanded by 500 terms and 10 phrases and resubmitted for retrieval (without the user ever having seen the first retrieval's results).

The differences between the details of that approach and this year's TREC 4 approach are due to two factors: the new document length normalization approach, and the much shorter queries. The normalization approach implies the number of expansion terms should be cut from 500 to, say, 100. The shorter queries imply that it may be easier to lose the focus of the expanded query on the original topic. So the expansion amount is reduced even further, to 50 single terms and 10 phrases, and the number of documents that the expansion is based upon is reduced to 20. The validity of this last assumption and reduction needs to be tested; but this hasn't been done yet.

Thus, the TREC 4 CrnlAE procedure is as follows: The new *Lnu* weighting scheme is used for the documents, with *ltu* weights for the original queries. These queries are initially run against the documents, retrieving the top 20 documents. These documents are marked relevant without examining them, and they and the original query are fed into the Rocchio relevance feedback process. The most frequently occurring 50 single terms and 10 phrases from the top 20 documents are added to query, and the new vector is weighted by

$$\begin{aligned} Q_{\text{new}} &= A * Q_{\text{old}} \\ &+ B * \text{average\_wt\_in\_rel\_docs} \\ &- C * \text{average\_wt\_nonrel\_docs} \end{aligned}$$

Run	Rec-Prec	Relevant-ret
lnc.ltc	1627 (-)	3210
lnc.ltc-Exp	2012 (+24%)	3634
Lnu.ltu	2326 (+43%)	3709
Lnu.ltu-Exp	2944 (+81%)	4350

Table 1: Comparisons of adhoc normalization and expansion

The *A.B.C* parameters of the Rocchio equation are set to 8.8.0. These parameters weight the original query terms higher than in standard relevance feedback, and disregard occurrences among the non-relevant documents.

The new query is re-run against the *Lnu* weighted documents, retrieving the final 1000 documents submitted as the official CrnlAE run.

Table 1 shows the effect of good normalization on the TREC 4 adhoc task. This is a very large increase for both the unexpanded and expanded cases. The increase is much larger than the 20% that had been observed in the development of the *Lnu* weighting scheme. We conjecture this is due to the much shorter queries of TREC 4, which are particularly unsuited for a cosine similarity function.

Our results compared to the other TREC 4 adhoc systems look very good, especially considering that the CrnlAE run is completely automatic, and a number of the other good runs involved substantial human involvement. We are best on only 2 of the 49 queries, but are above the median on 41 queries.

CrnlAL – Adhoc individual term locality run

Continuing in Cornell's tradition of presenting one development run, for which we have evidence it will work, and one experimental run, for which we hope it might work, we develop an entirely new similarity function, Individual Term Locality (ITL for short).

The overall ITL approach is like that of Cornell's local/global approaches of the past few years. An initial retrieval is done using global criteria, and the top retrieved documents are re-indexed and re-ranked using criteria based on matches within a certain locality or part of the documents. The difference is that using ITL, no attempt is made to break a document down into its component parts. Instead the entire document is repesented by a sequence of tuples such that for every location in the document for which there is a query term match, the term, location, and possible other information is kept. This is not a vector and if a query term occurs more than once in the document, it will occur more than once in the locality tuple list.

For every point in the document, we then estimate the point similarity of the text around that specific point to the query. The total ITL similarity of the document to the query is the maximum of the point similarities for all points within the document.

The point similarity is calculated by sorting the locality tuple list by increasing distance of each tuple from the point under consideration. The sorted list is gone through in increasing distance order, summing the contributions of the individual tuples. The tuple contribution is the product of several contributions, based upon the following factors:

1. Distance of the tuple from the point. The greater the distance, the less this tuple should contribute to whether the document is relevant around that point.
2. Number of times this tuple term has been seen before in this point similarity computation. A term occurring a second time shouldn't contribute as much as it did the first time it occurred (within this sorted distance tuple list).
3. Weight of the term in the query.
4. Certainty of term match. This is the constant 1.0 within the adhoc task, but in an erroneous environment such as OCR or speech retrieval, would be affected by the probability of this being a correct match.

Run	Best	≥ median	< median
CrnlAE	2	39	8
CrnlAL	3	36	10

Table 2: Comparative Ad-hoc results

Run	Recall-Precision	Total Rel Retrieved	R Precision	Precision 100 docs
CrnlAE	2944	4350	3384	3112
CrnlAL	2829	4297	3256	3002

Table 3: Ad-hoc results

5. Overall document length. A document length normalization effect that in all of our experiments so far can be completely ignored.
6. Relationship of this term to its immediately surrounding terms. A lot of different things can go into this factor. Some are
   (a) Whether surrounding terms occur closely together in the query.
   (b) Whether surrounding terms occur closely together in a set of relevant documents.
   (c) Syntactic relationship between terms (same noun phrase?)
   (d) Semantic relationship between terms

   For our experiments in the adhoc and routing tasks, we only considered the first two relationships (for the adhoc task, we worked in an initial retrieval environment, with the top documents being considered relevant for the purposes of the second relationship).

The benefit of ITL is that individual term occurrences are being considered. This is not a big benefit for ordinary statistical retrieval; Cornell and others have been looking at statistical local and passage matching for at least the past 8 years. However, it offers great potential for establishing a framework to bring in non-statistical information into a similarity computation. In particular, the certainty and linguistic factors above can have an influence at the individual term level. Thus ITL should be very useful for NLP, OCR, and retrieval of speech.

The present drawback of ITL is that it has not been established that the pure statistical factors of ITL can be used to rank documents as effectively as the normal global similarities. Once that can be shown, then incorporating the additional non-statistical information into the equation should improve retrieval.

The CrnlAL official adhoc run is a 3 pass run. The CrnlAE run is duplicated as the first two passes. The end result of the two passes for each query is an expanded query, a list of 20 presumed relevant documents, and a ranked list of 1750 documents. The presumed relevant documents are analyzed to determine the degree to which pairs of expanded query terms are statistically related to each other. Then each of the 1750 top documents are re-indexed relative to the expanded query, getting the occurrence location of each query term match. The ITL similarity function is calculated for each document by calculating the point similarity at each term match within the document, and taking the maximum point similarity over all term matches. The final similarity of the document was set to be the global similarity plus the ITL similarity.

The results are very good. CrnlAE was surprisingly very consistently above the median considering the short queries, and the possibility of losing the focus of the query if the initial retrieval did not perform well. We had expected the very good results on many queries to be partly counter-balanced by very poor results on others. The results for CrnlAL were also good, resulting in more "best" scores, but were less consistent.

The CrnlAL results are very encouraging, but for now say that the ITL approach using only statistical information is not quite as good as using the purely global statistical approach. The official average precision figure of .2829 is about 4% worse than the CrnlAE run. This agrees well with our preliminary tests on other TREC subcollections in which ITL by itself was 8% worse than the global similarity. This difference is noticable but not large. Given the fact that our global run improved anywhere from 12% to 45% this past year (depending on the measure), the ITL approach appears valid and ready for incorporation of non-statistical information.

Routing Experiments

The basis for our development routing experiment, CrnlRE, in TREC 4 is the same as in TREC 3, the relevance feedback approach of Rocchio [6, 10, 2]. The TREC 4 CrnlRE run starts with "Lnu" weighted documents and "ltu" weighted queries. Expressed in vector space terms, the final query vector is this initial query vector moved toward the centroid of the relevant documents, and away from the centroid of the non-relevant documents.

$$\begin{aligned} Q_{\text{new}} &= A * Q_{\text{old}} \\ &+ B * \text{average\_wt\_in\_rel\_docs} \\ &- C * \text{average\_wt\_nonrel\_docs} \end{aligned}$$

Terms that end up with negative weights are dropped (less than 3% of terms were dropped in the most massive query expansion below).

The CrnlRE run uses 64,64,2 as the A,B,C parameters in the above equation, and expands the query by adding the 50 single terms and 10 phrases that occur in the most relevant documents. This emphasizes the original query terms a bit more than in past TREC's and and reduces the expansion amount significantly. Experiments suggest that this change is an over-reaction to the new weighting approaches, and that more expansion terms weighted more heavily should be used. More details of these experiments will be given in the final version of this paper.

Unlike in TREC 3, after forming the feedback query, CrnlRE adds a separate stage of tweaking the query term weights even further on a per query basis. Dynamic Feedback Optimization, described in detail in SIGIR '95 [3], alters the weights by testing whether a mildly changed term weight performs better when run on the learning set of documents (those documents already seen and judged). If the changed weight performs better, then the new weight is kept; otherwise the weight reverts back to what it was originally. The six-pass DFO algorithm described in the SIGIR paper is run, with the DFO parameters as described in that paper. An individual term weight might be increased by about a factor of 5 over its original feedback weight.

The DFO modified CrnlRE queries are then ready to be run against the test set documents. The test set documents are weighted with the same "Lnu" scheme as used throughout this paper. The "Lnu" weights are independent of test collection statistics; the constant values for slope and pivot used in document normalization are the same as was used for the learning set. The final similarity is a simple inner product of the query and document weights, retrieving the top 1000 documents to be evaluated by NIST.

The second of the Cornell routing runs, CrnlRL, is a counterpart to the CrnlAL adhoc run. The same Rocchio procedure for forming a new feedback query is performed, except that only 10 terms and 2 phrases are added from the relevant documents. The DFO procedure is not performed for CrnlRL; the feedback query is ready to be run on the test set of documents immediately after the Rocchio procedure.

However, unlike CrnlRE, the actual running of the CrnlRL queries on the test documents is a two-pass procedure. The first pass calculates a global similarity using an inner product on the "Lnu" weighted documents. Then the second pass uses the ITL (Individual Term Locality) algorithm exactly as is used for the CrnlAL run. As well as attempting to increase precision by using a local match, it is to be hoped that all the information about tight clusters of coocurring terms derived from the learning set relevant documents can be taken advantage of.

Routing Results

Both CrnlRE and CrnlRL do reasonably but not spectacularly in comparison with other TREC 4 routing runs (Table 4). Average precision is above the median for the majority of the queries for both runs. (Table 5

Run	Best	≥ median	< median
CrnlRE	2	33	15
CrnlRL	0	30	20

Table 4: Comparative Routing Results

	Run	$X.Y$	$A.B.C$	R-prec	Total Rel	recall-prec
1.	CrnlRE	50.10	64.64.2	3658	4917	3380
2.	CrnlRL	10.2	64.64.2	3481	4789	3112

Table 5: Routing evaluation

gives the actual evaluation numbers for the two runs.

For Cornell, DFO does not seem to yield as large of a benefit on the TREC 4 routing task as it did on our TREC 2 and TREC 3 tasks that were reported on for the SIGIR conference. One problem may be that we seriously cut down on the amount of expansion being done, because we weren't sure of the interaction of our new document weighting schemes, the Rocchio algorithm, and DFO. This is probably a mistake since DFO should have ameliorated any poor interaction. Another potential problem is that some of the test set data differed markedly from the learning set data, and DFO may have not handled the new data well.

The CrnlRL run using ITL is an experimental run and not much is to be expected from it. The CrnlRL and CrnlAL official runs are actually the first time the procedures have ever been run after query expansion! Since we had no experience with expansion, we decided to severely limit the expansion for CrnlRL, which undoubtedly led to a lot of the differences between the CrnlRE and CrnlRL runs. We should do some additional runs and evaluations in this area.

It is clear we're not doing a good job taking into account the local term relationships derived from the learning set relevant documents. This information ended up having a very minimal effect on the final ITL similarity; it needs to be featured more prominently.

After the TREC conference, we did some analysis to determine how our results could be improved. Table 6 gives the evaluation of some of those runs. We should have been more trusting of our expansion techniques. Run 2 is the same run as the official CrnlRE run, except doubling the expansion of single terms from 50 to 100, and emphasizing the expanded terms a bit more by decreasing the importance ('A') of the original query weight.

However, the major improvement came as we increased the importance of terms occurring in non-relevant documents('C')! This was a surprise, since with our old weighting schemes like "ltc", the negative information had very little impact. Part of this is due to the length bias associated with cosine weighting. The relevant documents tend to be longer documents whose terms were down-weighted by cosine. With "Ltu" weighted documents in rocchio's formula, the ratio between the contribution of the non-relevant documents to the contribution of the relevant documents was much less (one fourth) as opposed to this ratio when "ltc" weighted documents are used. This is a substantial difference, and suggests that this ratio was being dominated by average length of documents rather than inherent worth of terms.

Once more accurate weights due to relevant and non-relevant document occurrences are obtained, we can decrease the importance of a term being in the original query, and add more expansion terms. Run 3

	Run	$X.Y$	$A.B.C$	R-prec	Total Rel	recall-prec
1.	CrnlRE	50.10	64.64.2	3658	4917	3380
2.	DFO-expmed	100.10	32.64.2	3865	5012	3512
3.	Exphigh	300.30	8.32.256	3724	5348	3489
4.	DFO-exphigh	300.30	8.32.256	4041	5410	3811
5.	CrnlRE-desc	50.10	64.64.2	3777	5111	3547
6.	DFO-high-desc	300.30	8.32.256	4145	5569	3977

Table 6: Post-TREC Analysis

Run	Best	≥ median	< median
CrnlSV(official)	0	22	3
CrnlSE	3	16	6

Table 7: Comparative Spanish Results

	Run	R-prec	Total Rel	recall-prec
1.	CrnlSV(official)	2801	1664	2234
2.	CrnlSE	3118	1748	2821

Table 8: Spanish evaluation

in Table 6 shows expansion by 330 terms and phrases, decreasing the importance of the original query, and increasing the importance of occurrences in the non-relevant documents. This run does not include the DFO optimization, and is already much better than the CrnlRE official run, retrieving an impressive 430 more relevant documents. Adding DFO in Run 4 gives an added improvement; not adding that many more relevant documents, but doing a substantially better job ranking them. This is as expected. The DFO algorithm optimizes performance only among the top documents.

The guidelines of TREC, when read carefully, forbid using certain sections of the text documents, namely those with manually added keywords (e.g. "DESCRIPT") This guideline is not always obeyed, especially by the new groups that are typically overwhelmed with all the other details of their first TREC. Runs 5 and 6 show some of the effects of including those fields. In those two runs, the manually indexed fields were included in the test documents, though not in the learning documents. As can be seen, there is a 5% to 6% improvement over the "legal" runs, which is reasonably substantial. Future TREC's should probably pay more attention to this restriction so that everybody is on an even footing.

Spanish

We did not do much at all for Spanish due to Professor Salton's final illness and then death on August 28th. The Spanish track due date of September 1 precluded any new efforts for Spanish. We attempted to run basically the same two runs as last year, except with our updated weighting approach. The pure vector run worked, but the Spanish expansion run counterpart to CrnlAE had a bug (we invoked the wrong parser, one that was not 8-bit clean). This was tracked down long after the deadline, so only the vector run is an official run, though we present both results.

The CrnlSV run is the pure inner product vector run. The query is weighted with "ltu" weights, and the documents with the "Lnu" weights presented earlier. The CrnlSE run starts off with the CrnlSV results as an initial run. The top 20 documents (without human intervention) are assumed to be relevant. The Rocchio relevance feedback procedure is invoked to expand the original query (50 terms are added), and reweight. This final query is then re-run against the documents, with the top 1000 documents being sent to NIST for evaluation.

SMART is very language independent. The main requirement is that there is some easy method of determining word boundaries, which is true for most, but not all, languages. The total human time for converting SMART to handle Spanish, fashion stemming rules, and forming a stop list was about 5-6 hours (done for TREC 3 [5]).

Spanish Ad-Hoc Results

Table 7 and Table 8 give results for both the official run CrnlSV and the unofficial run CrnlSE. Both did very well compared to the median of all groups doing Spanish, though there was a very significant gap between median and best individual query scores, suggesting one or two other groups did extremely well.

The vector run CrnlSV was above the median for 88% of the queries. The expansion run was above the median for fewer queries, but performed considerably better than CrnlSV for many of those queries. This is as expected; if the initial search is good enough so that the top 20 documents are, if not relevant, at least strongly related to relevant documents, then the reformulated query will work well. However, if the initial documents miss the topic area completely, then the expanded query will go off into the hinterlands, and do very poorly.

One Spanish-specific problem that we had intended to handle this year, but didn't have time for, is that accented characters occur inconsistently throughout the text. There are many cases where the same word occurs both with and without an accent on some letter. There are various transformations that can handle this, and we need to investigate which works well.

Another lack of our Spanish approach that we did not have time to correct, is that phrases are not considered. The SMART approach uses statistical phrases derived from frequently occurring pairs of adjacent non-stopwords. Since there is no linguistic base for the phrases, they can be derived as easily for Spanish as for English. Our English results suggest that an improvement of between 5% and 12% can be achieved when statistical phrases are added (the latter figure resulting from expansion by phrases).

Confusion

We have implemented a two-pass correction algorithm for the confusion track, which seeks to measure the retrieval degradation when the text is highly unreliable, for example with massive OCR errors.

One of the problems associated with traditional OCR correction algorithms is that they strongly depend on having a correct dictionary available, so that mangled words can be mapped onto likely correct words. This can lead to problems when

1. A dictionary is not available.
2. The dictionary coverage does not match the collection.
3. Proper nouns are important.

Our approach does not use a correct dictionary; it only uses the standard collection dictionary of the degraded text. This fits the standard SMART philosophy that the most reliable source of information about a large collection of text is the text itself. This frees SMART from being domain or even language dependent, and greatly reduces the human involvement of working with a new collection of text.

For the first pass, the queries are indexed using the collection dictionary (we assume that each query word occurs correctly at least once in the collection). The query is then expanded by adding all words in the collection dictionary that can be transformed into a query word with one transformation. Here, a transformation can be a deletion, insertion, or substitution of any alphanumeric character. Each one of these added query terms is weighted using an "idf" based upon both the correct query term's collection frequency, and the collection frequency of the transformed term. There are various restrictions on the transformation process, the most important one being a minimum length of the original query term (5 letters).

The TREC document text was degraded using random substitutions of characters, including blanks. The types of errors were therefore random rather than being systematic errors such as would be realistic in an OCR environment. Thus instead of only a few standard mis-scannings of a document word, there tended to be hundreds. Using our process a typical 20 term query is expanded to over 2000 terms, most of which occurred in very few documents!

We earlier experimented with allowing two tranformations per query term, and allowing a query term to be a prefix of a document term (in case the blank between the query term and the next term was deleted), but overall effectiveness was degraded in those preliminary tests. We need to re-examine these possibilities later.

The problems with this first pass approach is that document weights are not being accurately given. The word "antitrust" may occur in 10 different forms within a long document, and a match of each one of those forms is considered an important new piece of evidence. Our first pass retrieval results are thus dominated by long documents.

Run	Best	$\geq$ median	$<$ median
CrnlB	26	20	4
CrnlBc10	24	18	8

Table 9: Comparative Confusion Results

	Run	R-prec	Total Rel	recall-prec
1.	CrnlB	2415	1815	2084
2.	CrnlBc10	1769	1489	1431

Table 10: Confusion evaluation

To correct this, a second correction pass was implemented. The top retrieved documents at the end of the first pass are re-indexed, just as Cornell has been doing for years with our local-global runs. However, instead of using the collection dictionary for the re-indexing, a dictionary consisting of only query terms is used. Each word in a document is compared to this dictionary and is indexed by a query term only if zero or one transformations are required to exactly match the query term. Thus all single-transformation variations of a query term will map to that query term, and the problem of multiple forms of a term is resolved.

The documents are weighted, and a normal inner-product similarity function is applied, resulting in the final similarity value. Note that the document weighting scheme needs to be a slight variation of our normal scheme in that the number of unique tokens in a document can no longer be calculated. Instead, document length is normalized by the total number of tokens in the document.

For our official run, CrnlBc10, we submitted just the results of the first pass query expansion; we ran out of time and couldn't complete the second pass. We hope to have second pass results available by our workshop talk, though they are still not finished. At a later point, the 20% confusion level task will be performed.

Unfortunately, our base run, CrnlB, made on the correct version of the text, is not directly comparable to CrnlBc10. We made it at a time when we were still planning on incorporating stemming into our final confusion run. However, we did not complete (or even start) our stemming work. A fair base comparison would be a run on an unstemmed version of the correct collection, which would probably be 5% to 10% worse than CrnlB.

As Table 9 and Table 10 show, the base case and confusion do very well in comparison with other groups, though the actual level of performance is not that impressive. There are only 3 groups being compared at the 10% confusion level, which meant that an accurate comparison is not really possible. However, for both the base task and the 10% confusion task, our results are very good, giving the best result for roughly half of the queries.

This work is very much a preliminary investigation into the two pass correction approach. We haven't yet implemented a version handling stemming (though plural removal is handled automatically by the transformation algorithm). We expect the standard stemming expansion approach to work, adding any term that stems to the same stem as the query term. Then statistical phrases can be added to at least the second pass, the current approach being single term only. The next step after that is query expansion.

The combination of this two pass approach with the ILT approach outlined earlier offers all kinds of possibilities. A much deeper individual term match can be done (eg, more transformations) if the match can be weighted with the likelihood of the match being correct. If such certainty information is available directly from the OCR system, then it can be used.

Note that this combination correcting-2-pass,ILT approach can be used not only in an OCR environment, but in any retrieval situation where the data is known to be erroneous. The most intriguing possibility is that of speech retrieval, with the initial pass narrowing in on likely "documents" and the second correcting ILT pass using linguistic and other knowledge to decide if a match exists. The important consideration here is that this final analysis is done within the context of the query, which simplifies the task immensely.

Interactive Introduction

In our interactive track participation, we were interested in studying the effectiveness of interactive searching with minimal user intervention. In particular, we wanted to see if simple *relevance feedback* based query modification performs well as compared to a more involved query modification technique where the users can manually add/remove terms to/from the query. In a relevance feedback based interactive search, the users are only asked to judge the presented documents for relevance, a task much simpler than deciding what terms will be appropriate for a search.

We also wanted to test the effect of relevance judgments based on a deep (complete) reading of the presented documents as opposed to a shallow (quick) reading. We submitted two runs, *CrnlI1* – a run based on deep reading of the documents; and *CrnlI2* – a run based on shallow reading of the documents. An interactive search based on shallow reading of documents can be especially useful when the documents marked relevant in the search are used via relevance feedback for query modification and further retrieval of additional documents. It is possible that quick reading of a few documents is sufficient for generating a good search formulation for later use.

Interactive System Design

The searchers were instructed to find as many useful documents for a query as possible in (roughly) twenty minutes. For a query, an interactive session starts with the retrieval of ten documents using the initial query, and proceeds in iterations with obtaining relevance judgments for the ten documents, automatic query modification via relevance feedback, and retrieval of ten new documents using the modified query. Documents retrieved in each iteration are presented to the user in rank order. We implemented a simple, textual user interface with the following features:

- A user can browse a document by either moving **f**orward or **b**ackward.
- A user can mark a presented document **r**elevant, **n**on-relevant, or **c**an't decide.
- While viewing a document, a user can also decide to **q**uit the search or get **m**ore[1] documents, *i.e.*, go directly to the next iteration. All unseen documents (including the present one) are returned to the potentially retrievable pool.
- A user can also ask to look at the **t**itles from the current iteration. Even though this feature would be useful in systems where searchers are allowed to skip documents by looking at just the document titles, in our system, this feature was almost never used by the searchers. The utility of this feature was also diminished because all TREC documents don't have a well defined title.

One desirable feature that our system misses is the highlighting of the query terms in the documents presented to a searcher. This feature would speed up, and possibly improve the relevance assessment process.

Similar to our routing runs, we used Rocchio's formula to do relevance feedback (with parameters $\alpha = 8$, $\beta = 8$, and $\gamma = 4$). In every iteration, we added *ten* new terms and *two* new phrases to the current query, and reweighted the query according to Rocchio's formula. All documents marked relevant *up to this stage* in the search were used in relevance feedback. This ensures that the relevant documents from the last iteration do not drift the query away from relevance.

Interactive Evaluation

In the following, we refer to the person who conducted the search for us as the *searcher*, and we call the relevance assessor from NIST, the *assessor*.

Interactive Primary Task

[1] **m**ore was never used by the searchers in our TREC participation.

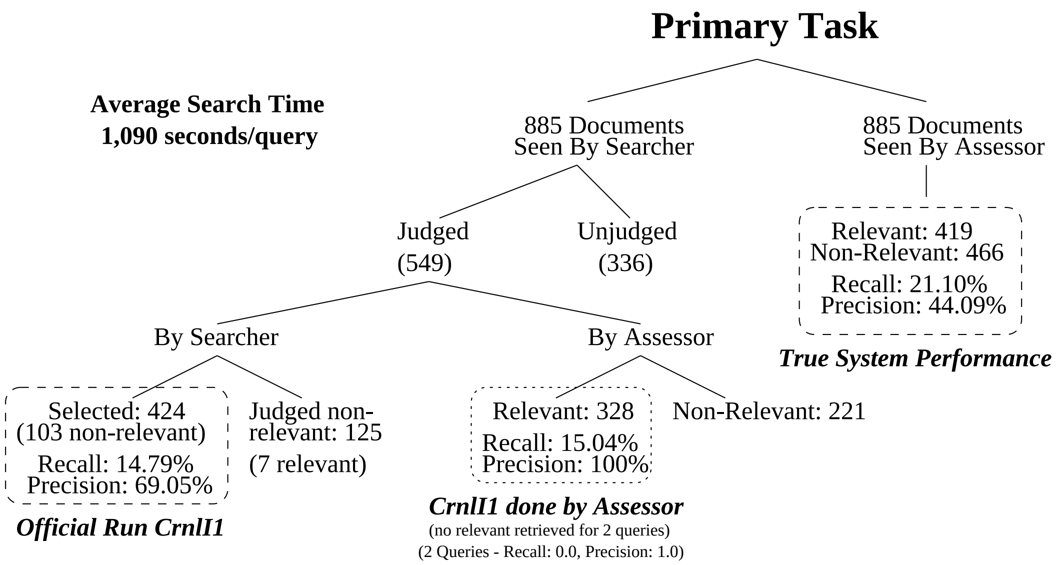

Figure 2: Evaluation of the primary interactive task.

Figure 2 shows the steps involved in official run *CrnlI1* for the primary interactive task. For the 25 queries used in the interactive track, the searcher looked at 885 documents in all. Out of these 885 documents, the searcher could not decide on the relevance for 336 documents, judging a total of 549 (885−336) documents. Out of the 549 documents judged, the searcher thought that 424 were relevant (and selected them for the official submission), and 125 were judged non-relevant. We immediately observe that our searcher's notion of relevance has a noticeable difference from the assessor's notion of relevance. Our searcher marked 103 documents as relevant that are actually non-relevant from the assessor's perspective. In effect, the *precision* of the official run is one measure of the *overlap between the searcher's and the assessor's notions of relevance*, and does not say much about the system performance.

The difference in recall between the assessor doing the search and the searcher doing the search is another measure of the difference in the searcher's and the assessor's view of relevance. We notice that out of the 125 documents that the searcher thought were non-relevant, 7 were actually relevant. Missing these 7 relevant documents marginally lowers the recall for *CrnlI1* (14.79% in place of a possible 15.04%). Of course, if the assessor was the real searcher, the search precision for the selected documents would be 100%. The actual system performance should be measured assuming that the real assessor will be judging *all seen documents* for relevance (there will be no unjudged documents), and that the recall and precision values will be based upon all seen documents as well (not only the ones judged relevant). Assuming similar time requirements in this scenario, the true system performance would have been R: 21.10% and P: 44.09%.

Our searcher marked 103 non-relevant documents as relevant, but missed only 7 relevant documents. In other words, our searcher was usually generous in assigning relevance – many non-relevant documents were selected and not too many relevant documents were missed. As the tendency of searchers to grant relevance can vary considerably from searcher to searcher, the recall and precision precision figures from the evaluation of the primary task are highly subjective to the person doing the search.

Interactive Secondary Task

The secondary task in the interactive track aims at measuring the effectiveness of the final search formulation generated after an interactive search. To generate a final query in the Smart system, we used all documents marked relevant by the searcher in an interactive session, and modified the *original query* via relevance feedback. We expanded the original query by *fifty* terms and *ten* phrases, and used Rocchio's feedback (with parameters $\alpha = 8$, $\beta = 8$, and $\gamma = 4$) for term weight modification. Freezing the documents *judged relevant* by the searcher at the top of the list, removing the documents judged non-relevant by the searcher,

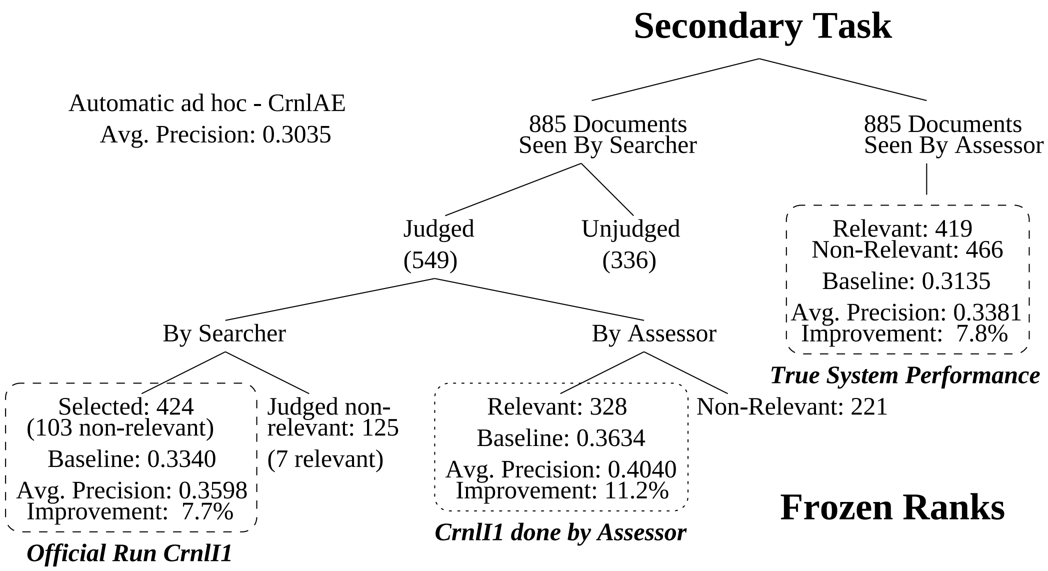

Figure 3: Evaluation of the secondary interactive task.

and returning the unjudged documents to the pool of potentially retrievable documents, we retrieved more documents using the modified query to get a total of 1,000 documents.

When only the documents judged relevant by a searcher are retained in the top ranks, evaluation based on rank freezing becomes highly sensitive to the overlap between the searcher's and the assessor's relevance judgments. Figure 3 shows that if the assessor was doing our official run *CrnlI1*, the average precision would have been 0.4040 in place of 0.3598. The better the overlap between the searcher's and the assessor's judgments, the higher is this figure. But we should observe that by freezing only the *selected documents* at the top, we also improve the baseline of our experiments. For example, if the documents selected by the searcher are frozen at the top, and the base query (from our ad hoc automatic run *CrnlAE*) is used to retrieve more documents until we have 1,000 documents, the average search precision increases from 0.3035 (*CrnlAE*) to 0.3340. If the documents selected by the assessor (actually relevant) are frozen at the top and the base query used to retrieve additional documents, the average precision improves further to 0.3634.

Using rank freezing for only the selected documents, this dependence of average precision on searcher's notion of relevance hinders a direct comparison between the average precision for various participants. A better evaluation of this task would have been the traditional rank freezing evaluation where all seen documents (relevant or non-relevant) are frozen at the top. Using such an evaluation, the true average precision for the run *CrnlI1* is 0.3381.

We evaluated our official run *CrnlI1* by residual collection analysis as well. As the documents retrieved during different iterations depend upon the relevance assessments from the previous iterations, we note that the set of the seen documents can differ if the searcher was running our system, as opposed to if the assessor was running it. To obtain a uniform residual collection for our analysis, we remove any documents that were viewed by a user (searcher/assessor) for a query. Now we can directly compare

1. the base query (from *CrnlAE*),
2. the query generated using the documents marked relevant by the searcher,
3. the query generated if the assessor was marking the documents presented to the searcher (to study the effect of overlap between the searcher's and the assessor's relevance judgments), and
4. the query generated if an assessor was running our system.

	Base (*CrnlAE*)	*CrnlI1*	*CrnlI1* Assessor's Judgments	Assessor Running Smart
Avg. Precision Improvement	0.1302	0.1605 +23.2%	0.1657 +27.3%	0.1681 +29.1%
Exact R-Precision	0.1867	0.2153	0.2185	0.2270

Table 11: Residual collection evaluation of the secondary interactive task.

Run	Avg. Time per Query (seconds)	25 Queries				Average Recall (Micro)	Average Precision (Micro)	Modified Queries Full Coll.
		Total Seen	Total Judged	Judged Relevant	Judged Non-Rel.			
Deep Run *CrnlI1*	1,090	885	549	424 (103 non-rel.)	125 (7 rel.)	0.1479	0.6905	0.3609
Shallow Run *CrnlI2*	854	1,540	970	506 (144 non-rel.)	464 (63 rel.)	0.1378	0.5941	0.3424 −5.1%

Table 12: Deep run vs. shallow run.

Using residual collection analysis, the results in Table 11 show that the difference in the quality of the final search formulation is marginal (.1605 vs. .1681) irrespective of who did the relevance assessments. This fact was not at all apparent from the rank freezing analysis, and suggests the evaluation methodology of future experiments needs to re-considered.

Interactive Deep Run vs. Shallow Run

One of our interactive runs was based on shallow (quick) reading of the documents presented to the searcher. We wanted to test if weak relevance assessments for more documents would result in a better final query as compared to judging fewer documents with in-depth reading of the documents. Results from the deep run (*CrnlI1*) and the shallow run (*CrnlI2*) are compared in Table 12. In the shallow run, the searcher was able to look at many more documents in less time, but the quality of the relevance assessments was not as good as the deep run (average precision fell from 0.6905 to 0.5941).

When more documents are selected (506 in place of 424), one expects the recall to go up. But, surprisingly, we observe a fall in average recall for the shallow run as opposed to the deep run. On further analysis, we find that the average recall is heavily influenced by queries that have very few relevant documents. For example, if we consider only the queries that have at least 25 relevant documents (21 queries), the average recall for the deep run (0.1342) is actually less than the average recall for the shallow run (0.1420). We have observed that such queries, which are generally "hard" for a keyword based retrieval system, oftentimes have relevance buried in few sentences within a relevant document. Such relevance is more amenable to discovery during a deep reading of a document. Therefore, the deep run does much better on such queries.

As the main aim of this experiment was to study the improvement in query quality as a result of a deep/shallow run, we modified the original query, once using all documents judged relevant in the deep run, and then using all documents judged relevant in the shallow run. We searched the *entire collection*[2] using these modified queries. The non-interpolated average precisions from this experiment are shown in the last column of Table 12. We observe that the shallow run has slightly poor performance (5.1% worse) than the deep run, but with the searcher spending less time on a query. Once again, if we remove the queries with fewer than 25 relevant documents, the average precisions for the deep and the shallow run become 0.3675 and 0.3632, respectively. Now the shallow run is almost at par with the deep run (just 1.2% worse).

These results show that if the aim of an interactive session is to improve the quality of the search

[2] The residual collections for the two runs are different.

formulation, a shallow reading of several documents can work almost as well as a deep reading of a few documents, taking less time. We believe that with a better user interface, in particular by highlighting the initial query terms, the quality of the shallow run would benefit more than the deep run, and it is possible that a shallow run might outperform a deep run. We will explore this aspect in our future experiments.

Interactive Note

We discovered a minor mistake in our official interactive track submission. As the search formulation changes over iterations, for some queries our system generated document similarities in later iterations which were greater than the numerical similarity of documents from the previous iterations. As the TREC evaluation programs sort the retrieved documents by similarity, some documents retrieved in the later iterations were placed before some documents retrieved in earlier iterations. This affected the ranking for Q_0 in the secondary task. The primary task was unaffected due to its *set oriented* evaluation. The ranking for documents retrieved in the batch mode was also unaffected. For this reason, the official results for our secondary interactive task are slightly different than the real results.

Run	Official Average Precision	Real Average Precision
CrnlI1	0.3589	0.3598
CrnlI2	0.3225	0.3243

Efficiency

Efficiency issues are becoming increasingly important in these TREC experiments as retrieval methods become more complicated and expensive. Thus it is important to have at least some discussion of efficiency within a paper like this.

SMART is a reasonably fast system. It indexes documents at a rate of about 600–800 megabytes per hour (including inverted files) on a low-end Sun Sparc (Model 20-51). Simple vector retrieval runs can be quite fast. The CrnlVS vector run takes less than a second for all 25 queries combined, when asked to retrieve 10 documents per query. Keeping track of the top 1000 documents is currently much more expensive, adding about 1.3 seconds per query.

The more complicated approaches are much more time consuming, ranging up to 4 minutes per query (CrnlAL). The CrnlAL time for each query includes 3 retrieval passes, re-indexing and relevance feedback expansion of 20 documents, complete re-indexing of 1750 top documents in preparation for the ITL pass, and approximately 50 similarity computations for each of those 1750 documents (one for each query term match within the document).

Luckily, in actual practice the execution times of the complicated methods can be cut down drastically. The massive query expansion approaches will benefit greatly from optimization efforts such as those discussed in our TREC 1 work. Some of the effectiveness increase of the massive query expansion will have to be traded back in order to get reasonable efficiency, but the results of TREC 1 show the effectiveness cost will not be prohibitive. The new approaches like ITL have had no attention paid to optimization, but obviously the efficiency can be improved once they have been proven to be effective.

Comparison with past TREC's

It is difficult to determine how much systems are improving from TREC to TREC since the queries and the documents are changing. For example, in TREC 3 the "Concept" field of the queries was removed. These terms proved to be very good terms for retrieval effectiveness in TREC 1 and TREC 2; thus the TREC 3 task without them is a harder task than previous TRECs. The TREC 4 task was even more difficult since so much more of the text was removed from the queries. This makes the TREC 4 results much more realistic for the ad-hoc retrieval, since most users will type in a sentence at most, but it also depresses the results. To get a handle on how much SMART has improved in the past three years, Table 13 presents the results of running our TREC 1-4 systems on both the TREC 3 and TREC 4 ad-hoc tasks. The automatic approach of

Methodology	Run	TREC 3 Task Rec-Prec	TREC 4 Task Rec-Prec
TREC 1	ntc.ntc	2067 (-)	1538 (-)
TREC 2	lnc.ltc	2842 (+38%)	1627 (+6%)
TREC 3	lnc.ltc-Exp	3419 (+65%)	2012 (+31%)
TREC 4	Lnu.ltu-Exp	3852 (+86%)	2944 (+91%)

Table 13: Comparisons of past approaches with present

SMART has been improving at an average rate of almost 25% per year so far. This rate will probably start tailing off in the future, especially if the queries remain short, but we still expect substantial improvements for next year!

Conclusion

The Cornell SMART Project is again a very active participant in this year's TREC program. With the exception of our interactive track participation, everything we have presented here is completely automatic and uses no outside knowledge base (other than a small list of stopwords to ignore while indexing). Manual aids to the user can be built on top of this system to provide even greater effectiveness.

Our investigations into document length normalization show that the cosine normalization used by SMART for past TRECs is not well suited for full text documents. The "Lnu" weighting scheme presented here handles the TREC documents with much less of a length bias. This is very important for the short TREC 4 type of queries, which are strongly affected by normalization issues.

Our ad hoc expansion approach, exemplified by the CrnlAE run, works very well. Queries are expanded by terms occurring in the top retrieved documents, and reweighted using the Rocchio relevance feedback formula.

Our Individual Term Locality (ITL) similarity approach attempts to come up with a new similarity function operating on individual term occurrences instead of the normal vector representation of the document. This holds great potential for development in the future, since non-statistical information about individual terms can easily be used within the model.

Our routing run this year performed unspectacularly (right about the median); we haven't been able to analyze why. The Dynamic Feedback Optimization approach used very successfully in our experiments for SIGIR did not perform well here.

Our interactive results showed that minimal involvement by users, just having users judge the relevance of documents, can result in a very effective retrieval set.

We tried a new 2-pass dictionaryless correction algorithm for the confusion (OCR) track. All "correction transformations" were made between terms occuring in the erroneous documents, and the query terms, no correct dictionary was involved. This performed very well, at least as well as the other entries in the track (CrnlBc10 had the best results on half the queries), even though only the first half of the algorithm was actually run.

The Spanish results were again considerably above the median, and used no language knowledge. The runs were exactly the same runs as we run on the English tasks.

A comparison with previous years' TREC approaches show that SMART is averaging about a 25% improvement per year. That rate will be difficult to maintain, but we'll try!

References

[1] Chris Buckley. *Massive Query Expansion for Relevance Feedback.* Cornell University, 1995.

[2] Chris Buckley, James Allan, and Gerard Salton. Automatic routing and ad-hoc retrieval using SMART : TREC 2. In D. K. Harman, editor, *Proceedings of the Second Text REtrieval Conference (TREC-2)*, pages 45–56. NIST Special Publication 500-215, March 1994.

[3] Chris Buckley and Gerard Salton. Optimization of relevance feedback weights. In Ed Fox, Peter Ingwersen, and Raya Fidel, editors, *Proceedings of the Eighteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 351–357, New York, July 1995. ACM.

[4] Chris Buckley, Gerard Salton, and James Allan. Automatic retrieval with locality information using SMART. In D. K. Harman, editor, *Proceedings of the First Text REtrieval Conference (TREC-1)*, pages 59–72. NIST Special Publication 500-207, March 1993.

[5] Chris Buckley, Gerard Salton, James Allan, and Amit Singhal. Automatic query expansion using SMART : TREC 3. In D. K. Harman, editor, *Proceedings of the Third Text REtrieval Conference (TREC-3)*. NIST Special Publication 500-225, 1995.

[6] J.J. Rocchio. Relevance feedback in information retrieval. In Gerard Salton, editor, *The SMART Retrieval System—Experiments in Automatic Document Processing*, chapter 14. Prentice Hall, Englewood Cliffs, NJ, 1971.

[7] Gerard Salton. *Automatic Text Processing — the Transformation, Analysis and Retrieval of Information by Computer*. Addison-Wesley Publishing Co., Reading, MA, 1989.

[8] Gerard Salton. Developments in automatic text retrieval. *Science*, 253:974–980, August 1991.

[9] Gerard Salton and Chris Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[10] Gerard Salton and Chris Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41(4):288–297, 1990.

[11] Amit Singhal, Gerard Salton, Mandar Mitra, and Chris Buckley. Document length normalization. Technical Report TR95-1529, Cornell University, 1995.

Appendix:Interactive system description

0.1 Experimental Conditions

1. **Searcher Characteristics**
   (a) Number of searchers in experiment: 2
   (b) Number of searchers per topic: 2
   (c) Age of searchers: 25 and 27
   (d) IR searching experience of searchers: None
   (e) Educational level of searchers: Ph.D. Students
   (f) Undergraduate major of searchers: Computer Science
   (g) Experience/familiarity with subject of topic: Variable (per topic)
   (h) Work affiliation of searchers: Department of Computer Science, Cornell University

2. **Task Description:** Find as many documents as possible that (you think) are relevant to a topic in approximately *twenty minutes.*

3. **Training**
   (a) Description of the training process: In the training phase, the searchers used the search interface to conduct searches for old TREC topics (selected topics from TREC topics 1–200).
   (b) Time for training: Searcher 1 - (approx.) 240 minutes, Searcher 2 - (approx.) 195 minutes.

0.2 Search Process

1. **Clock Time**

	Seconds per Topic			
Run	Mean	Median	Std. Dev.	Range
CrnlI1	1,090	1,051	220	649 – 1,469
CrnlI2	854	859	180	471 – 1,165

2. **Documents Viewed**
   (a) **Viewing:** A document is considered viewed if any part of it, *other than the title,* is presented to a searcher.
   (b) Number of documents viewed:

	Documents per Topic			
Run	Mean	Median	Std. Dev.	Range
CrnlI1	35.40	33	9.76	19 – 53
CrnlI2	61.60	66	15.22	30 – 90

3. **Iterations**
   (a) **Iteration:** Retrieve and display ten documents using the query from the previous iteration (initial query used in the first iteration), obtain relevance judgments for these documents, and modify query using these relevance judgments.

   The searcher can end any iteration by asking to do another iteration or by quiting the search, without looking at all ten documents. In this case, the unseen documents are returned to the potentially retrievable pool. Even if the searcher looks at just one document presented in an iteration, we consider it a full iteration.

   (b) Number of iterations:

	Iterations per Topic			
Run	Mean	Median	Std. Dev.	Range
CrnlI1	4	4	1.08	2 – 6
CrnlI2	6.32	7	1.63	3 – 9

4. **Number of Search Terms**

   We consider a single term or a phrase as a *term*. For example, topic 203 has five terms initially:

   - Single terms: **tire**, **econom**, **impact**, **recycl**
   - Phrase: **econom impact**

   (a) Number of terms in initial query:

	Initial Terms per Topic			
Run	Mean	Median	Std. Dev.	Range
CrnlI1	9.36	8	4.18	3 – 17
CrnlI2	Same as *CrnlI1*			

   (b) Number of terms in final query:

   To obtain a final query we added fifty new single terms and ten new phrases to the *initial query* via relevance feedback, based upon the the searcher's relevance assessments.

	Final Terms per Topic			
Run	Mean	Median	Std. Dev.	Range
CrnlI1	69.36	68	4.18	63 – 77
CrnlI2	Same as *CrnlI1*			

5. **System Features**

   The following system features were used by the searchers whenever needed:

   f Forward: Browse document forward one page (35 lines)

   b Back: Browse document backward one page (35 lines)

   y/r Rel: Mark document relevant

   n Non-Rel: Mark document non-relevant

   c No Clue: Mark document can't decide

   t Titles: Show titles from current iteration (seldom used)

   m More: Get more documents by going to next iteration directly (never used)

   q Quit: Quit search

6. **Errors**

   Not applicable.

7. **Narrative for Topic 236**

   (a) **Set-Up**

   The search was conducted using 2 windows — one displayed the query text, and was adjusted such that only a single query could be displayed in it at a time; the actual search was conducted in the other window.

   The windows looked typically like the following:

   - **Search Window:**

```
Smart (ntq?): Trec 236
Num  Action Sim Title
875162     19.51 COMMANDERS SAY WOMEN SHOULDN'T BE IN COMBAT FEMALE OFFICERS PRE
543651     19.21 With PM-Welfare Overhaul Bjt</HEAD> WORK</HEAD> CHILD CARE</HEA
875966     18.61 NEW FORMULA JACKS UP COST OF RENEWING LICENSE TAGS </HEADLINE>
560047     18.02
754212     17.87 Business, Administration Support National Product Liability Law
845980     17.79 GOVERNOR SIGNS BILL LIMITING TOBACCO-SAMPLE GIVEAWAYS </HEADLIN
778338     17.75 Congressman Wants More Regulations For Cosmetics</HEAD>
576988     17.29 U.S. Votes Against Law Of Sea Convention</HEAD>
558522     16.99 Wetlands Threatened by Water-Level Increases</HEAD>
857354     16.79 A WILD RIDE OFF THE CANADA COAST </HEADLINE>
Hit return to continue ...
```

- **Query Window:**

```
<num> Number:  236

<desc> Description:

Are current laws of the sea uniform?  If not, what
are some of the areas of disagreement?

</top>

<top>

<num> Number:  237
```

(b) **Search Process**

The search proceeds thus:

i. At the system prompt we enter "**Trec 236**". The system logs the current time in a log file as:

```
Trace: entering trec_pager
Elapsed Time:   34387.345579
```

and retrieves 10 documents in response to query 236 and displays them in the search window as shown above. While the system is retrieving documents, we read the query-text. Since the queries are very short, this takes very little time.

ii. On typing return, the text of the top-ranked document is displayed by the pager. The screen now looks like this [3]:

```
857354     16.79 A WILD RIDE OFF THE CANADA COAST </HEADLINE>
Hit return to continue ...
.- --- 875162 /fsys/thor/k/trec.d3/sjm/sjm_158 977136 984193

.n 13 39
 SJMN91-06171051 </DOCNO>

.w 76 207
  Chart, photo; PHOTO: Associated Press; Maj. Christine Prewitt, right, and Lt.
```

[3] We show successive screens with an overlap of 2 lines so that the reader may easily follow the sequence of actions.

```
Brenda Marie Holdener testify (color).  </CAPTION>

.s 305 734
  The Pentagon's senior officers insisted Tuesday that current rules barring
women from serving in combat should continue, while some members of the Senate
Armed Services Committee heartily endorsed allowing women to fly warplanes in
combat.;    The crowded hearing marked the first formal congressional inquiry
into allowing women to fly combat aircraft since the Persian Gulf war, where
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@ f:Forward, b:Back, y/r:Rel, n:Non-Rel, c:No Clue, t:Titles, m:More, q:Quit
```

f and **b** are used to move a page forward or backward in the document text. When we have read enough of the document to be able to judge its relevance, we type the appropriate letter (**y**/**n**/**c**). Note that, we avoid reading the entire document only when relevance is clearly established before the end of the document. **c** is used when we are unable to decide the document's relevance.

iii. When a relevance judgement is entered, the pager automatically displays the next document. Thus, when the first document has been viewed (we were unable to decide whether this particular document was relevant or not), and the appropriate judgement (**c**) entered, the screen looks like this:

```
.w 6498 6526
<CITY>  Washington  </CITY>
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@ f:Forward, b:Back, y/r:Rel, n:Non-Rel, c:No Clue, t:Titles, m:More, q:Quit c
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
.- --- 543651 /fsys/thor/k/trec.d2/ap/ap880617 154899 158945

.n 13 37
 AP880617-0048 </DOCNO>

.t 191 227
With PM-Welfare Overhaul Bjt</HEAD>

.w 271 559

   Here is a comparison of the current welfare
system and the welfare overhaul bills passed by the House and
Senate. Negotiators will reconcile differences in the two bills
before sending a measure to the White House, where it is not clear
whether President Reagan will sign it.
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@ f:Forward, b:Back, y/r:Rel, n:Non-Rel, c:No Clue, t:Titles, m:More, q:Quit
```

iv. This continues until the last document retrieved in an iteration has been judged. At this stage, the system internally modifies the query using relevance feedback and returns 10 more documents and displays them as before. The whole process (steps (ii) to (iv)) is repeated for the new iteration.

v. Finally, when we exhaust the allotted time for a query or feel that we have examined a sufficient number of documents, we press **q**, and return to the system prompt. The system logs the current time again to mark the end of the search:

```
Trace: leaving trec_pager
Elapsed Time:   35615.352744
```

The documents retrieved after the first iteration were:

```
Num  Action Sim Title
807542     2.24 Conference Considers Ban on Toxic Ocean Dumping; US Wants More
808752     2.20 Global Accord Reached to Clean Up World's Oceans</HEAD>
590029     2.16 U.S. Territorial Waters Extended From Three to 12 Miles</HEAD>
766572     2.10 U.S. Safety Investigators Want More Access To Cruise Ships</HEA
598342     2.08
517530     2.03 Twenty-Three Nations Sign Treaty to Combat Terrorism at Sea</HE
575157     1.98 U.S.-Soviet Negotiators Call For Halt To Overfishing</HEAD>
779976     1.97 US-Soviet Accord on Bering Sea Producing No Bonanzas</HEAD>
605081     1.95
515744     1.90 U.S. Law To Close PLO Mission Assailed By General Assembly</HEA
Hit return to continue ...
```

The documents retrieved after the second iteration were:

```
Num  Action Sim Title
795682     4.10 Child Rights Convention Comes Into Force</HEAD>
753624     3.94 UN Panel Approves Measure Opposed by U.S., Panama, Others</HEAD
529790     3.92 World Court Rules Against Washington Over PLO Office</HEAD> <NO
538572     3.90 With PM-Summit-Reagan Bjt</HEAD> Reagan, Gorbachev Expected To
588328     3.84 U.N. Conference Adopts Convention Against Illicit Drug Traffick
521133     3.84 U.N. General Assembly Slaps United States For PLO Eviction Atte
811205     3.84 U.S. Negotiators to Push Limited Mining Moratorium</HEAD>
575767     3.84 U.S. Citizens Group Says U.S.-U.N. Relations in Disarray</HEAD>
590177     3.81 U.S. Official Says New 12-Mile Offshore Limit Will Help Deter S
794140     3.81 Experts Say Iraq's Actions Against Embassies Violate Internatio
Hit return to continue ...
```

Thus, for this query, we ran 2 iterations after the initial retrieval, and examined 29 of the 30 documents retrieved. 10 documents were judged non-relevant, 6 were deemed relevant, and we were unable to decide 13 documents. All this took 1228.007165 seconds (20.47 mins.).